

**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Aktenzeichen:

102 54 658.4

Anmeldetag:

22. November 2002

Anmelder/Inhaber:

Philips Intellectual Property & Standards GmbH,
Hamburg/DE

(vormals: Philips Corporate Intellectual Property
GmbH)

Bezeichnung:

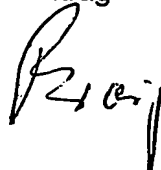
Mikrocontroller und zugeordnetes Verfahren zum
Abarbeiten der Programmierung des Mikrocontrollers

IPC:

G 06 F 12/14

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ur-
sprünglichen Unterlagen dieser Patentanmeldung.

München, den 23. September 2003
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag


Brosig

BEST AVAILABLE COPY



BESCHREIBUNG

Mikrocontroller und zugeordnetes Verfahren zum Abarbeiten der Programmierung des Mikrocontrollers

Die vorliegende Erfindung betrifft einen Mikrocontroller, dessen Programmierung in
5 mindestens einer maschinenabhängigen Assemblersprache vorgenommen ist, bei der die
Assemblerbefehle mit Ausnahme bedingter Programmsprünge bzw. bedingter Pro-
grammverzweigungen im wesentlichen unabhängig von Daten ausführbar sind, wobei
10 - im Falle einer erfüllten Sprung- bzw. Verzweigungsbedingung, beispielsweise
mindestens eines erfüllten Statusflags, mindestens ein Programmzähler mit einer
neuen Adresse und/oder mit einem neuen Wert beaufschlagbar ist und
- im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung, beispie-
weise mindestens eines nicht erfüllten Statusflags, die Instruktion beendet wird.

Die vorliegende Erfindung betrifft des weiteren ein Verfahren zum Abarbeiten der in
15 mindestens einer maschinenabhängigen Assemblersprache vorgenommenen Programmie-
rung eines Mikrocontrollers der vorgenannten Art.

Als Mikrocontroller werden Einchip-Mikrocomputer bezeichnet, die in der Regel zum
Steuern von Geräten verwendet werden und bei denen C[entral]P[rocessing]U[nit],
20 Speicher und Ports auf einem Chip integriert sind. Die Programmierung von Mikrocont-
rollern erfolgt in maschinenabhängiger Assemblersprache. Bei den bekannten Assembler-
sprachen werden hierbei mit Ausnahme der bedingten Programmsprünge bzw. der be-
dingten Programmverzweigungen alle Assemblerbefehle unabhängig von Daten ausge-
führt.

25 Ein bedingter Programmsprung bzw. eine bedingte Programmverzweigung wird üblicher-
weise wie folgt realisiert: Die zu überprüfende Bedingung, in der Regel mindestens ein
Statusflag, wird getestet. Wird hierbei festgestellt, dass ein Sprung bzw. eine Verzweigung

stattfinden soll, so wird der Programmcounter mit einer neuen Programmadresse (= mit einem neuen "Wert") geladen. Wenn kein Sprung bzw. keine Verzweigung stattfinden soll, so wird die Instruktion beendet, denn der Programmcounter enthält ja schon automatisch den nächsten Wert, das heißt die nächste Adresse.

5

Ein derartiges Procedere bedingt, dass bei bedingten Programmsprüngen bzw. bei bedingten Programmverzweigungen ein zeitlicher Unterschied in der Instruktionsausführung auftreten kann. Dieser zeitliche Unterschied in der Instruktionsausführung liegt darin begründet, dass im springenden bzw. verzweigenden Falle zusätzlich noch der

10 Programmcounter auf einen neuen Wert (auf eine neue Programmadresse) gesetzt wird, während im nicht-springenden bzw. nicht-verzweigenden Falle die Instruktion nach dem Bedingungstest beendet ist.

15 Dies bedeutet, dass die Befehlsausführung bedingter Sprünge bzw. bedingter Verzweigungen in Mikrocontrollerprogrammen üblicherweise unterschiedliche Ausführungszeiten und damit auch unterschiedliche, anhand dynamischer Strommessungen ermittelbare Stromwerte aufweist, und zwar in Abhängigkeit davon, ob ein bedingter Sprung bzw. eine bedingte Verzweigung ausgeführt wird oder nicht.

20 Eine gängige, auch einen Missbrauch durch Angreifer ermöglichende Methode zur Softwareanalyse, zum Beispiel zum Ermitteln von kryptographischen Schlüsseln, besteht darin, mittels einer speziellen Timinganalyse bedingte Programmsprünge bzw. bedingte Programmverzweigungen zu identifizieren und anhand des identifizierten Programmablaufs Rückschlüsse auf die verarbeiteten Daten zu gewinnen.

25

Damit lassen sich allein mittels des zeitlichen Ablaufs der bedingten Sprunginstruktion bzw. der bedingten Verzweigungsinstruktion Rückschlüsse auf die in dieser Instruktion getesteten Daten ziehen, was beispielsweise bei einem unbefugten Angriff auf besonders sicherheitssensible Abschnitte eines Mikrocontrollerprogramms, wie etwa auf einen

30 kryptographischen Schlüssel, von extremem Nachteil ist.

Ausgehend von den vorstehend dargelegten Nachteilen und Unzulänglichkeiten sowie unter Würdigung des umrissenen Standes der Technik liegt der vorliegenden Erfindung die Aufgabe zugrunde, einen Mikrocontroller der eingangs genannten Art sowie ein Verfahren der eingangs genannten Art so weiterzuentwickeln, dass nach außen hin unsichtbar ist, ob bei einem bedingten Programmsprung bzw. bei einer bedingten Programmverzweigung tatsächlich gesprungen bzw. verzweigt wird oder nicht.

Diese Aufgabe wird durch einen Mikrocontroller mit den im Anspruch 1 angegebenen Merkmalen sowie durch ein Verfahren mit den im Anspruch 5 angegebenen Merkmalen gelöst. Vorteilhafte Ausgestaltungen und zweckmäßige Weiterbildungen der vorliegenden Erfindung sind in den jeweiligen Unteransprüchen gekennzeichnet.

Mithin ist die Lehre der vorliegenden Erfindung in einer gesicherten Operation von Mikrocontrollern, insbesondere von Smart Card-Controllern, bei bedingten Programmsprüngen bzw. bei bedingten Programmverzweigungen zu sehen.

Hierzu wird der interne Ablauf der Instruktionsabarbeitung des bedingten Sprungs bzw. der bedingten Verzweigung erfindungsgemäß wie folgt modifiziert: Im Falle eines Sprungs bzw. einer Verzweigung wird der dem Mikrocontroller zugeordnete Programmcounter (nachstehend auch als Programmzähler bezeichnet) - wie an sich bekannt - mit einem neuen Wert geladen. Im nicht-springenden bzw. nicht-verzweigenden Falle wird hingegen nunmehr anstelle eines Beendens der Sprung- bzw. Verzweigungsinstruktion ebenfalls der Programmcounter, insbesondere unter Einbeziehung mindestens einer additionalen Logik, neu geladen, allerdings diesmal mit seinem eigenen Wert.

Die Vorgehensweise gemäß der vorliegenden Erfindung bedeutet mit anderen Worten, dass das Ergebnis der Testbedingung nicht mehr dazu verwendet wird, die interne Programmabarbeitung zu beenden oder nicht; vielmehr wird das Ergebnis der Testbedingung in bevorzugter Weise zum Ansteuern mindestens eines Multiplexers eingesetzt, der in Abhängigkeit vom Testergebnis entweder eine neue Adresse oder den Programmcounterausgang zum Speichern an den Programmcountereingang legen kann.

Demzufolge wird der Programmcounter in jedem Falle mit einer neuen Adresse, das heißt mit einem neuen Wert geladen, und zwar unabhängig davon, ob gesprungen bzw. verzweigt werden soll oder nicht. Daraus ergibt sich ein identisches zeitliches Ablaufverhalten für beide Fälle.

5
Gemäß einer besonders erfinderischen Weiterbildung ergibt sich eine weitere Verbesserung des Unsichtbarmachens von bedingten Sprüngen bzw. von bedingten Verzweigungen dann, wenn sowohl das Testen der Sprung- bzw. Verzweigungsbedingung als auch das Laden des Programmcounters mit komplementären Daten ausgeführt werden (= sogenanntes "current blinding" durch komplementären Programmcounter), denn dann
10 kann eine den Mikrocontroller angreifende Person anhand dynamischer Strommessungen nicht mehr unterscheiden, ob gesprungen bzw. verzweigt wurde oder nicht.

In einer zweckmäßigen Ausführungsform der vorliegenden Erfindung lässt sich der Ablauf bedingter Programmsprünge bzw. bedingter Programmverzweigungen dergestalt
15 optimieren, dass die Abarbeitung des bedingten Sprungs bzw. der bedingten Verzweigung wahlweise auf die vorbeschriebene Weise (Programmcounter wird stets neu geladen) oder auf die an sich bekannte Weise (= Nicht-Sprung bzw. Nicht-Verzweigung beendet Instruktion) vollzogen wird. Die Steuerung dieser Option oder Wahlmöglichkeit
20 erfolgt durch mindestens ein spezielles Bit (= sogenanntes "select bit").

Die vorbeschriebene Option oder Wahlmöglichkeit kann in vorteilhafter Weise für folgende Zwecke genutzt werden:

- (i) in unkritischen Teilen der Programmierung des Mikrocontrollers lässt sich der
25 durch das Laden des Programmcounters bedingte Verlust an Performance (--> längere Ausführungszeit im Falle eines Nicht-Sprungs bzw. einer Nicht-Verzweigung) unterdrücken, wenn die select bit-Option auf die übliche Abarbeitung eingestellt ist;
- (ii) wenn die select bit-Option in beliebigen Reihenfolgen ein- und ausgeschaltet
30 wird, zum Beispiel per Randomfunktion oder aber auch mit anderen geeigneten

Bitfolgen, werden alle Nicht-Sprünge bzw. Nicht-Verzweigungen mal als "kurze" Ausführungszeit und mal als "lange" Ausführungszeit wahrgenommen; hierdurch wird eine Analyse der Daten per Instruktionsausführungszeit von bedingten Sprüngen bzw. von bedingten Verzweigungen deutlich erschwert, so dass ein Angreifer durch die unterschiedlichen Ausführungszeiten bei identischen Daten im Falle des Nicht-Sprungs bzw. der Nicht-Verzweigung einer bedingten Instruktion bewusst getäuscht und in die Irre geführt wird.

Zusammenfassend sind bei der vorliegenden Erfindung also erhebliche Vorteile

- 10 - in den erheblich erschwerten Möglichkeiten der Analyse der Daten bei bedingten Sprüngen;
- in der identischen Ausführungszeit bei bedingten Sprüngen durch unbedingtes Nachladen des Programmcounters; und/oder
- 15 - in der frei wählbaren Variation, ob bei Nicht-Sprung bzw. bei Nicht-Verzweigung eine kurze Befehlsausführungszeit oder eine lange Befehlsausführungszeit stattfinden soll,

zu sehen. Mithin führt die vorliegende Erfindung unabhängig von der Struktur des (Mikrocontroller-)Programms stets zu denselben dynamischen Stromwerten und verhindert somit ein missbräuchliches sowie unerlaubtes Ausforschen zeitbedingter dynamischer Stromanalysen.

Die vorliegende Erfindung betrifft schließlich ein elektrisches oder elektronisches Gerät, gesteuert mittels mindestens eines Mikrocontrollers der vorstehend dargelegten Art. Wie bereits vorstehend erörtert, gibt es verschiedene Möglichkeiten, die Lehre der vorliegenden Erfindung in vorteilhafter Weise auszugestalten und weiterzubilden. Hierzu wird einerseits auf die dem Anspruch 1 sowie dem Anspruch 5 nachgeordneten Ansprüche verwiesen, andererseits werden weitere Ausgestaltungen, Merkmale und Vorteile der vorliegenden Erfindung nachstehend anhand des durch Figur 1 veranschaulichten Ausführungsbeispiels näher erläutert.

Es zeigt:

Fig. 1 in schematischer Darstellung ein Blockschaltbild eines Ausführungsbeispiels eines mit dem Verfahren gemäß der vorliegenden Erfindung betriebenen Mikrocontrollers gemäß der vorliegenden Erfindung.

In Figur 1 ist ein Ausführungsbeispiel für einen als SmartCard-Controller ausgebildeten Mikrocontroller 100 dargestellt, dessen Programmierung in einer maschinenabhängigen Assemblersprache vorgenommen ist und abgearbeitet wird. Hierbei werden verfahrensgemäß die Assemblerbefehle mit Ausnahme bedingter Programmsprünge bzw. bedingter Programmverzweigungen unabhängig von Daten ausgeführt.

Im Falle einer erfüllten Sprung- bzw. Verzweigungsbedingung, etwa eines erfüllten Statusflags, wird ein dem Mikrocontroller 100 zugeordneter Programmzähler 10 mit einer neuen Adresse und/oder mit einem neuen Wert geladen; die Besonderheit des Mikrocontrollers 100 ist nun darin zu sehen, das bei diesem Mikrocontroller 100 im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung, etwa eines nicht erfüllten Statusflags, die Instruktion nicht notwendigerweise beendet wird, sondern dass der Programmzähler 10 in diesem Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung anstelle des Beendens der Instruktion wahlweise mit seinem bisherigen Wert neu geladen werden kann.

Hierzu weist der Mikrocontroller 100 eine mittels des Ergebnisses des Testens der Sprung- bzw. Verzweigungsbedingung ansteuerbare Multiplexeinheit 20 auf, wobei an den Eingang des Programmzählers 10

- im Falle einer erfüllten Sprung- bzw. Verzweigungsbedingung die neue Adresse und/oder der neue Wert und
- im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung die Adresse am Ausgang des Programmzählers 10 und/oder der Wert am Ausgang des Programmzählers 10 gelegt wird.

Mithin wird das tatsächliche Ergebnis der Testbedingung nicht mehr dazu benutzt, um die interne Programmabarbeitung zu beenden oder nicht; vielmehr wird das Ergebnis der Testbedingung zum Ansteuern des Multiplexers 20 eingesetzt, der in Abhängigkeit vom Testergebnis entweder eine neue Adresse (nämlich bei erfüllter Sprung- bzw. Verzweigungsbedingung) oder den Ausgang des Programmcounters 10 (nämlich bei nicht-erfüllter Sprung- bzw. Verzweigungsbedingung) zum Speichern an den Eingang des Programmcounters 10 legen kann.

Demzufolge wird der Programmcounter 10 in jedem Falle mit einer neuen Adresse, das heißt mit einem neuen Wert geladen, und zwar unabhängig davon, ob gesprungen bzw. verzweigt werden soll oder nicht. Daraus ergibt sich ein identisches zeitliches Ablaufverhalten für beide Fälle, so dass das Verfahren im Mikrocontroller 100 gemäß Figur 1 unabhängig von der Struktur des (Mikrocontroller-)Programms stets zu denselben dynamischen Stromwerten führt und demzufolge ein missbräuchliches sowie unerlaubtes Ausforschen zeitbedingter dynamischer Stromanalysen verhindert wird.

Eine weitere Verbesserung des Unsichtbarmachens von bedingten Sprüngen bzw. von bedingten Verzweigungen ergibt sich, indem sowohl das Testen der Sprung- bzw. Verzweigungsbedingung als auch das Laden des Programmcounters 10 mit komplementären Daten ausgeführt werden (= sogenanntes "current blinding" durch komplementären Programmcounter), denn dann kann eine den Mikrocontroller 100 angreifende Person anhand dynamischer Strommessungen nicht mehr unterscheiden, ob gesprungen bzw. verzweigt wurde oder nicht.

Bei der vorliegenden Erfindung gemäß Figur 1 lässt sich der Ablauf bedingter Programmsprünge bzw. bedingter Programmverzweigungen dergestalt optimieren, dass die Abarbeitung des bedingten Sprungs bzw. der bedingten Verzweigung wahlweise auf die vorbeschriebene Weise (Programmcounter 10 wird stets neu geladen) oder auf die an sich bekannte Weise (= Nicht-Sprung bzw. Nicht-Verzweigung beendet Instruktion) vollzogen wird. Die Steuerung dieser Option oder Wahlmöglichkeit erfolgt durch ein spezielles Bit (= sogenanntes "select bit");

Die vorbeschriebene Option oder Wahlmöglichkeit kann für folgende Zwecke genutzt werden:

- (i) in unkritischen Teilen der Programmierung des Mikrocontrollers 100 lässt sich der durch das Laden des Programmcounters 10 bedingte Verlust an Performance (--> längere Ausführungszeit im Falle eines Nicht-Sprungs bzw. einer Nicht-Verzweigung) unterdrücken, wenn die select bit-Option auf die übliche Abarbeitung eingestellt ist;
- (ii) wenn die select bit-Option in beliebigen Reihenfolgen ein- und ausgeschaltet wird, zum Beispiel per Randomfunktion oder aber auch mit anderen geeigneten Bitfolgen, werden alle Nicht-Sprünge bzw. Nicht-Verzweigungen mal als "kurze" Ausführungszeit und mal als "lange" Ausführungszeit wahrgenommen; hierdurch wird eine Analyse der Daten per Instruktionsausführungszeit von bedingten Sprüngen bzw. von bedingten Verzweigungen deutlich erschwert, so dass ein Angreifer durch die unterschiedlichen Ausführungszeiten bei identischen Daten im Falle des Nicht-Sprungs bzw. der Nicht-Verzweigung einer bedingten Instruktion bewusst getäuscht und in die Irre geführt wird.

BEZUGSZEICHENLISTE

- | | |
|------|--|
| 100 | Mikrocontroller, insbesondere SmartCard-Controller |
| 10 | Programmcounter oder Programmzähler |
| 5 20 | Multiplexeinheit oder Multiplexer |

PATENTANSPRÜCHE

1. Mikrocontroller (100), dessen Programmierung in mindestens einer maschinenabhängigen Assemblersprache vorgenommen ist, bei der die Assemblerbefehle mit Ausnahme bedingter Programmsprünge bzw. bedingter Programmverzweigungen im wesentlichen unabhängig von Daten ausführbar sind, wobei
- 5 - im Falle einer erfüllten Sprung- bzw. Verzweigungsbedingung, beispielsweise mindestens eines erfüllten Statusflags, mindestens ein Programmzähler (10) mit einer neuen Adresse und/oder mit einem neuen Wert beaufschlagbar ist und
 - im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung, beispielsweise mindestens eines nicht erfüllten Statusflags, die Instruktion beendet wird,
- 10 dadurch gekennzeichnet,
dass der Programmzähler (10) im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung anstelle des Beendens der Instruktion mit seiner bisherigen Adresse und/oder mit seinem bisherigen Wert wahlweise neu beaufschlagbar ist.
- 15 2. Mikrocontroller gemäß Anspruch 1,
gekennzeichnet durch
mindestens eine mittels des Ergebnisses des Testens der Sprung- bzw. Verzweigungsbedingung ansteuerbare Multiplexeinheit (20), wobei der Eingang des Programmzählers (10)
- im Falle einer erfüllten Sprung- bzw. Verzweigungsbedingung mit der neuen
 - 20 Adresse und/oder mit dem neuen Wert und
 - im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung mit der Adresse am Ausgang des Programmzählers (10) und/oder mit dem Wert am Ausgang des Programmzählers (10)
- beaufschlagbar ist.

3. Mikrocontroller gemäß Anspruch 1 oder 2,
gekennzeichnet durch
eine Ausgestaltung als SmartCard-Controller.

5 4. Elektrisches oder elektronisches Gerät, gesteuert mittels mindestens eines Mikro-
controllers (100) gemäß mindestens einem der Ansprüche 1 bis 3.

5. Verfahren zum Abarbeiten der in mindestens einer maschinenabhängigen Assembler-
sprache vorgenommenen Programmierung eines Mikrocontrollers (100), wobei

- 10 - die Assemblerbefehle mit Ausnahme bedingter Programmsprünge bzw. bedingter
Programmverzweigungen im wesentlichen unabhängig von Daten ausgeführt werden,
- im Falle einer erfüllten Sprung- bzw. Verzweigungsbedingung, beispielsweise
mindestens eines erfüllten Statusflags, mindestens ein Programmzähler (10) mit
einer neuen Adresse und/oder mit einem neuen Wert geladen wird und
15 - im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung, beispiels-
weise mindestens eines nicht erfüllten Statusflags, die Instruktion beendet wird,

dadurch gekennzeichnet,

- dass der Programmzähler (10) im Falle einer nicht erfüllten Sprung- bzw. Verzweigungs-
bedingung anstelle des Beendens der Instruktion mit seiner bisherigen Adresse und/oder
20 mit seinem bisherigen Wert wahlweise neu geladen wird.

6. Verfahren gemäß Anspruch 5,

dadurch gekennzeichnet,

dass an den Eingang des Programmzählers (10)

- 25 - im Falle einer erfüllten Sprung- bzw. Verzweigungsbedingung die neue Adresse
und/oder der neue Wert und
- im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung die Adresse
am Ausgang des Programmzählers (10) und/oder der Wert am Ausgang des
Program mzählers (10)

30 gelegt wird.

7. Verfahren gemäß Anspruch 5 oder 6,

dadurch gekennzeichnet,

dass das Testen der Sprung- bzw. Verzweigungsbedingung und/oder das Laden des

5 Programnzählers (10) mit komplementären Daten ausgeführt wird.

8. Verfahren gemäß mindestens einem der Ansprüche 5 bis 7,

dadurch gekennzeichnet,

dass im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung die Option

10 zwischen dem Beenden der Instruktion und dem Neu-Beaufschlagen des Programm-
zählers (10) mit seiner bisherigen Adresse und/oder mit seinem bisherigen Wert durch
mindestens ein spezielles Bit (sogenanntes "select bit") gesteuert wird.

9. Verfahren gemäß mindestens einem der Ansprüche 5 bis 8,

15 dadurch gekennzeichnet,

dass im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung in unkritischen
und/oder sicherheitstechnisch nicht sensiblen Programmteilen die Option des Beendens
der Instruktion insbesondere durch das spezielle Bit gewählt wird.

20 10. Verfahren gemäß Anspruch 8 oder 9,

dadurch gekennzeichnet,

dass die spezielle Bit-Option in beliebiger Reihenfolge ein- und ausgeschaltet werden
kann, beispielsweise mittels mindestens einer Randomfunktion und/oder mittels
mindestens einer geeigneten Bitfolge.

25

ZUSAMMENFASSUNG

Mikrocontroller und zugeordnetes Verfahren zum Abarbeiten der Programmierung des Mikrocontrollers

- Um einen Mikrocontroller (100), dessen Programmierung in mindestens einer maschinen-
5 abhängigen Assemblersprache vorgenommen ist, bei der die Assemblerbefehle mit Ausnahme bedingter Programmsprünge bzw. bedingter Programmverzweigungen im wesentlichen unabhängig von Daten ausführbar sind, wobei
- im Falle einer erfüllten Sprung- bzw. Verzweigungsbedingung, beispielsweise mindestens eines erfüllten Statusflags, mindestens ein Programmzähler (10) mit
10 einer neuen Adresse und/oder mit einem neuen Wert beaufschlagbar ist und
 - im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung, beispielsweise mindestens eines nicht erfüllten Statusflags, die Instruktion beendet wird,
- sowie ein Verfahren zum Abarbeiten der in mindestens einer maschinenabhängigen Assemblersprache vorgenommenen Programmierung des Mikrocontrollers (100) so
15 weiterzuentwickeln, dass nach außen hin unsichtbar ist, ob bei einem bedingten Programmsprung bzw. bei einer bedingten Programmverzweigung tatsächlich gesprungen bzw. verzweigt wird oder nicht, wird vorgeschlagen, dass der Programmzähler (10) im Falle einer nicht erfüllten Sprung- bzw. Verzweigungsbedingung anstelle des Beendens der Instruktion mit seiner bisherigen Adresse und/oder mit seinem bisherigen Wert wahlweise neu
20 beaufschlagbar ist.

Fig. 1

100 ↗

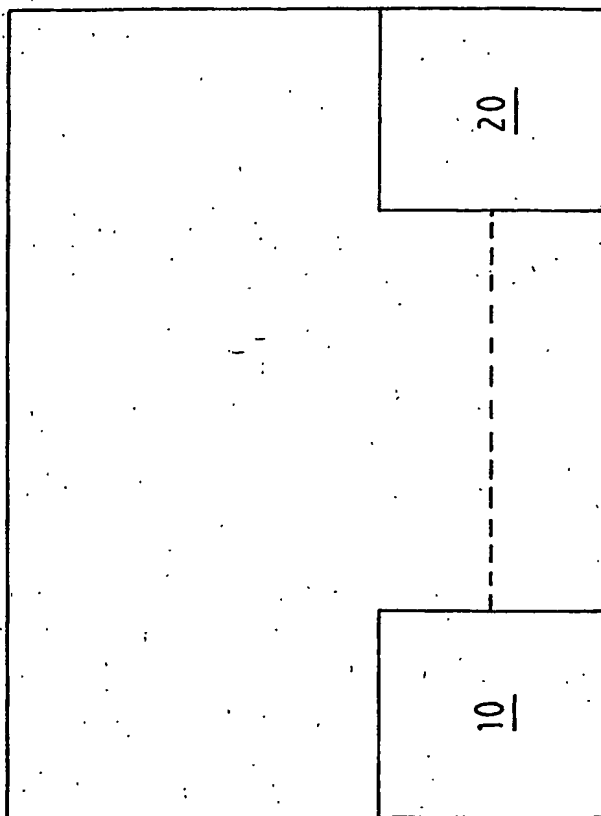


Fig.1

100

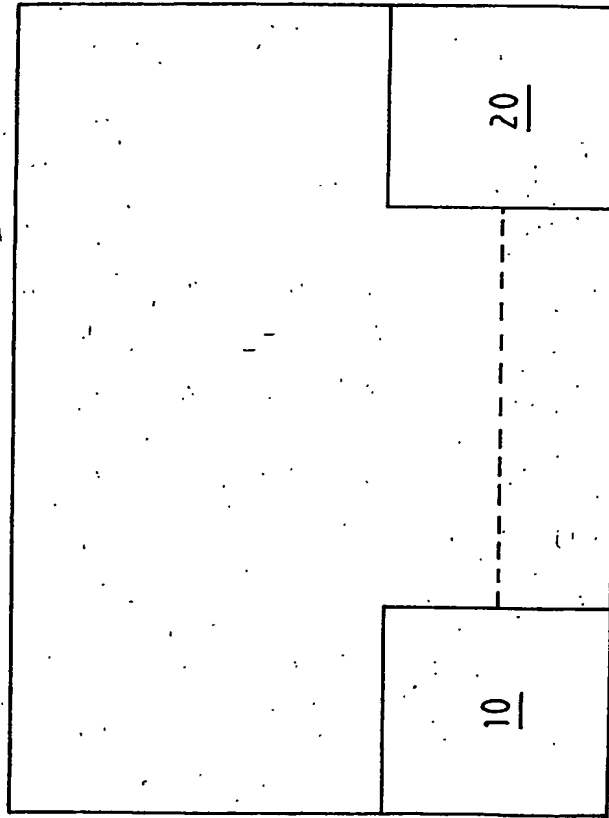


Fig.1

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.